Neurocomputing 326–327 (2019) 3–14

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

On the use of evolutionary time series analysis for segmenting paleoclimate data

M. Pérez-Ortiz[a,1,*], A.M. Durán-Rosal[b], P.A. Gutiérrez[b], J. Sánchez-Monedero[a], A. Nikolaou[c], F. Fernández-Navarro[a], C. Hervás-Martínez[b]

[a] Department of Quantitative Methods, Universidad Loyola Andalucía, Third Building, Córdoba 14004, Spain
[b] Department of Computer Science and Numerical Analysis, University of Córdoba, Rabanales Campus, Albert Einstein building, Córdoba 14071, Spain
[c] German Aerospace Center Institute of Planetary Research Planetary Physics Rutherfordstraße 2 12489 Berlin

ARTICLE INFO

Article history:
Received 20 November 2015
Revised 9 June 2016
Accepted 1 November 2016
Available online 18 September 2017

Keywords:
Time series segmentation
Genetic algorithms
Clustering
Paleoclimate data
Tipping points
Abrupt climate change

ABSTRACT

Recent studies propose that different dynamical systems, such as climate, ecological and financial systems, among others, present critical transition points named to as *tipping points* (TPs). Climate TPs can severely affect millions of lives on Earth so that an active scientific community is working on finding early warning signals. This paper deals with the development of a time series segmentation algorithm for paleoclimate data in order to find segments sharing common statistical patterns. The proposed algorithm uses a clustering-based approach for evaluating the solutions and six statistical features, most of which have been previously considered in the detection of early warning signals in paleoclimate TPs. Due to the limitations of classical statistical methods, we propose the use of a genetic algorithm to automatically segment the series, together with a method to compare the segmentations. The final segments provided by the algorithm are used to construct a prediction model, whose promising results show the importance of segmentation for improving the understanding of a time series.

© 2017 Published by Elsevier B.V.

1. Introduction

In contrast to the famous statement of Linnaeus "*natura non facit saltus*" (or nature makes no leaps), it has been proven that some points of no return, thresholds and phase changes are widespread in nature and these are often non linear [1]. Such events can be rarely anticipated and some of them can have detrimental consequences on Earth's climate and large-scale impacts on human and ecological systems. This increases the imperious necessity of studying, analysing and developing techniques for characterising them in order to construct reliable early warning systems. Although the human being have influenced their local environment for millennia, e.g. reducing biodiversity, it is now, since the industrial revolution, that truly global changes are being noticed [2,3]. Examples that are currently receiving attention include the potential collapse of the Atlantic thermohaline circulation, the dieback of the Amazon rainforest or the decay of the Greenland ice sheet [1]. Formally, a climate "tipping point" (TP, also known as "little things can make a big difference") occurs when a small change in forcing triggers a strongly nonlinear response in the internal dynamics of part of the climate system, qualitatively changing its future state.

The critical relevance of early TPs detection has produced a growing attention of the scientific community. Lenton differentiates between several types of TPs, and presents some indicators that can help to detect them, such as the increase of autocorrelation of the series values [4]. In [5], more concrete techniques regarding data processing and indicators are presented. They study a bank of methods using only simulated ecological data, concluding in concordance with the literature that there is no unique best indicator for identifying an upcoming transition. They also conclude that all the methods require specific data-treatment. Up to our knowledge, all previous works tackle the TP detection with statistical methods trying to select (by trial and error) the method (and the time-window) most suitable to detect those transitions. They require an intensive data preprocessing that includes, for instance, the use of Gaussian filters or rolling windows that introduce extra parameters (such as the width of the Gaussian function or size of the window) that need to be optimised [4,5]. The main limitation behind these methods is that different TPs require specific treatment, which is the specific objective that this paper tries to tackle.

Although one of the main areas of research for time series is their modelling [6], time series segmentation is emerging as a very interesting field, aiming to provide a compact representation of the

* Corresponding author.
E-mail address: i82perom@uco.es (M. Pérez-Ortiz).
[1] This paper has been invited to be included in the "Special Issue Neurocomputing-HAIS2014".

https://doi.org/10.1016/j.neucom.2016.11.101
0925-2312/© 2017 Published by Elsevier B.V.

time series values, dividing it into segments and using an abstract representation of each segment. It is very important for time series representation and time series mining [7–9] and is commonly used as a pre-processing step for different mining tasks [9–12] (e.g. clustering, classification or motif detection) and for data compressing [13,14]. In this way, segmentation algorithms have been used in many different fields, such as paleoecological problems [15], phoneme recognition [16] or paleontological climate [17]. Some recent works have proposed the use of algebraic segmentation for the specific case of short-term [18,19] and short nonstationary time series [20].

In this context, this paper deals with climate time series segmentation to effectively analyse Dansgaard–Oeschger (DO) events (i.e. the consequence of different tipping points occurred in the past). The main objective of the paper is to improve the ability to understand paleo-records and identify abrupt climate changes (in this case, DO events). TPs have been studied using predefined and fixed time-windows, or specifying different windows for each one (as it has been suggested that there are different types of TPs, represented by different time spans and statistical features). In other words, it is well-known that there exist nonlinear triggers (known as TPs) which anticipate abrupt climate changes, but the way to characterise and detect these triggers is not clearly defined, mainly because the nature of each TP is usually different from the rest (they occur in different time spans and there exist different statistical features that help to detect them). Motivated by this fact, the main aim for using a segmentation algorithm is to automatically detect and exploit these differences between the TPs. The combined use of an evolutionary segmentation algorithm, a clustering method and a set of features previously linked to TP detection would ideally serve two different objectives: (1) group similar patterns according to the set of features (which should help to detect TPs) and (2) find the best time span to detect each TP (this search is guided by a measure of the compactness of the clusters obtained). Note that, given the nature of both the evolutionary algorithm and the fitness function, these two objectives are achieved jointly, optimising the segmentation so that the differentiation between the clusters is maximal (according to a set of predefined features, which would guide the search towards clusters composed of TPs against clusters of characteristics radically different). We believe that this tool could help the expert to study and analyse the nature of TPs and to construct a diagnostic–predictive model to anticipate this type of events. We introduce an evolutionary time series segmentation method as a first step to better understand the paleoclimate data used. This segmentation provides a more compact representation of the time series by splitting it into segments with similar statistical behaviour [8]. This segmentation analysis avoids the necessity of specifying predefined sliding windows for the different TPs, which is one of the main difficulties of previous TP detection methods [5]. Furthermore, the segmentation algorithm is able to detect and exploit differences between the TPs (via the clustering phase), which could be interesting for the prediction phase. We address the segmentation problem as a heuristic search problem with the proposal of a Genetic Algorithm (GA) to overcome the limitations of traditional statistical methods. The GA segments the data trying to obtain diverse clusters of segments based on six statistical properties (for more information about time series clustering see [21]). These statistics have been selected given their potential for the discrimination of paleoclimate TPs. The fitness of each solution is estimated using different cluster validity indexes that have been highlighted for their good behaviour in the literature. However, measuring the quality of a segmentation can be only achieved by expert evaluation of the solutions given by the algorithm. An important contribution of this paper is a quantitative method to perform comparisons with respect to an expected ideal segmentation of the series to assess the robustness and stability of the method. This method allows the evaluation of an unsupervised segmentation algorithm (as the one developed) with a minimal effort by the expert, who has only to provide the ideal segmentation. We test the proposal with two of the most studied paleoclimate proxy data series, which demonstrates tipping behaviour using the oxygen isotope δ^{18}O of the Greenland ice cores [22,23] and includes climate records from −60,000 years to the present. Finally, our experiments also demonstrate the feasibility of training a predictive machine learning model as an early warning signal, using the statistics computed for the resulting segments and the values of the time series.

A preliminary study in this direction was presented in [17], where an evolutionary segmentation algorithm was applied to analyse the segments preceding TPs, using the quotient of the number of segments divided by the sum of squared error of the clustering as the fitness function. This paper extends the previous work with the following contributions:

- The evaluation of the segmentations obtained by the different versions of the algorithm has been automatised using two different metrics, the former comparing the segmentation to an ideal segmentation and the latter evaluating the stability of the algorithm. These evaluation metrics are used as a post-evolution analysis of the TP detection capability of each version of the algorithm.
- This automatic evaluation method has been used to perform a battery of experiments comparing the results obtained by a total of 10 different cluster validity indexes, used as fitness functions. The new fitness functions improve the results of [17] to a great extent. These metrics, which measure cluster compression, are used to guide the algorithm to different solutions, which are later compared.
- The best segmentation obtained has been used to devise a classification model able to recognise abrupt climate changes based on the characteristics of the previous segment. This model can be used to better understand the characteristics of potential early warning signals of TPs.
- Some improvements have been included in the algorithm, such as the binary coding (which alleviates the computational load of the algorithm) and a restriction of the minimum segment size.

The rest of the paper is organised as follows. Section 2 presents the segmentation algorithm, while Section 3 presents a proposal for segmentation comparison. Section 4 discusses the experimental results and the last section depicts the conclusions and future research.

2. Segmentation algorithm

Given a univariate time series $Y = \{y_n\}_{n=1}^{N}$, our objective is to divide the values of y_n into m consecutive subsets or segments. These segments should present a homogeneous behaviour regarding the values of y_n. This is done by partitioning the time indexes $(n = 1, \ldots, N)$ into segments: $s_1 = \{y_1, \ldots, y_{t_1}\}, s_2 = \{y_{t_1}, \ldots, y_{t_2}\}, \ldots, s_m = \{y_{t_{m-1}}, \ldots, y_N\}$, where t's are the cut points and are subscripted in ascending order $(t_1 < t_2 < t_{m-1})$. The cut points belong to two segments (the one before and the one after to analyse consistently the transition from one segment to the next). The integer m and the cut points $t_i, i = 1, \ldots, m-1$, are the parameters to be determined by the algorithm. As done in [24], we extend this setting by trying to group the segments into k different classes or clusters $(k < m)$, where k is a parameter defined by the user. In this way, each segment, s_l, will be associated to a class label: $(s_1, z_1), \ldots, (s_m, z_m)$, where z_l, $l = 1, \ldots, m$, is the class label of the lth segment and takes values in a set of k different labels, $z_l \in \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$.

Evolutionary time series segmentation algorithm:

Require: Time series.
Ensure: Best segmentation of the time series.
1: Generate a random population of P time series segmentation solutions.
2: Evaluate all segmentations of the initial population by using a set of statistics, a clustering algorithm and a predefined fitness function f.
3: **while not** Stop Condition **do**
4: Store a copy of the best solution.
5: Select parents from current population.
6: Generate offspring: apply crossover and mutation to construct new candidate segmentations.
7: Evaluate the fitness of the offspring.
8: Merge parents and offspring.
9: Replace current population with selected segmentations of the previous union.
10: **end while**
11: Compare with the ideal segmentation.
12: **return** Resulting segmentation.

Fig. 1. Main steps for the GA. Note that in this case the stop condition is controlled by a parameter g associated to the number of generations of the evolutionary algorithm.

2.1. Summary of the algorithm

The Genetic Algorithm considered in this paper can be included in the area of time series segmentation [8,10–12,14,18,25–27]. Our main objective is to devise an unsupervised methodology to identify time segments with similar statistical behaviour [24]. This is a necessary step to study the characteristics of these time segments and be able to analyse the temporal transitions between different states (i.e., types of segments) or to construct a prediction model with a dynamic-window [17].

In the algorithm, each possible segmentation is represented as an array of binary values (chromosome representation), where a value of 1 represents a cut point. The evolution starts from a population of randomly generated segmentations. Mutation and crossover operators are applied to explore and exploit the search space. This procedure is repeated during g generations. To evaluate a solution (or segmentation), we select a set of statistics to be calculated for each segment. Then, similar segments are grouped using a clustering process. A pseudocode of the segmentation algorithm is included in Fig. 1. A detailed description of each step of the GA are defined in the following sections.

2.2. Chromosome representation

As stated before, each individual chromosome consists of an array of binary values, where the length of the chromosome is the time series length, N. Each position c_i stores whether the time index t_i of the time series represents a cut point for the evaluated solution.[2] In this sense, for a given segment s_i delimited by the cut points t_{i-1} and t_i ($t_{i-1} < t_i$), the corresponding chromosome values will be $c_{i-1} = 1,\ c_i = 1$ and $c_l = 0, \forall l | t_{i-1} < l < t_i$.

2.3. Initial population

In order to initialise the population of P segmentations solutions, an average segment length has to be specified by the user ($\overline{sl}$). Taking into account that the cut points belong to two segments, the number of cut points will be $\overline{m} = \lceil \frac{N}{\overline{sl}-1} \rceil$, so the chromosomes are binary arrays where $\overline{m}$ random positions are 1s and the rest are 0s. In order to check that time segments are not overly small we include the constraint that no consecutive cut points are allowed (setting then a minimum length of 3 points per segment).

[2] Note that the first and last points of the chromosome are considered as cut points.

2.4. Fitness evaluation

As stated before, the main objective of the algorithm is not only to segment the time series but also to find time patterns with similar behaviour (whether they represent TPs or not). Because of this, a clustering process is applied to group similar segments. The evaluation of the quality of a segmentation solution consists of three different steps: extracting the characteristics of the segments, applying a clustering process and measuring the quality of this clustering.

2.4.1. Extracting segment characteristics

Given that the segments in a chromosome can have different length, an approach is designed to project all the segments into the same dimensional space. Six statistical metrics are considered and measured for all chromosome segments (note that each segment corresponds to an univariate time series). Then, the similarities between segments can be calculated in the 6-dimensional space. Consider s_s as a segment fulfilling the previously stated conditions (i.e., s_s is a segment delimited by the cut points t_{s-1} and t_s, where the segment length is $t_s - t_{s-1} + 1$). The mapping is done by the function $f : \mathbb{R}^{(t_s - t_{s-1}+1)} \rightarrow \mathbb{R}^6$, in the following way:

$$f(s_s) = \left(S_s^2, \gamma_{1s}, \gamma_{2s}, a_s, MSE_s, AC_s\right) \tag{1}$$

where the different characteristics are defined as:

1. Variance (S_s^2): It measures the variability of the segment:

$$S_s^2 = \frac{1}{t_s - t_{s-1} + 1} \sum_{i=t_{s-1}}^{t_s} (y_i - \overline{y_s})^2, \tag{2}$$

where y_i are the time series values of the segment, and $\overline{y_s}$ is the average value of the segment.

2. Skewness (γ_{1s}): It represents the (vertical) asymmetry of the distribution of the series values in the segment with respect to the arithmetic mean:

$$\gamma_{1s} = \frac{\frac{1}{t_s - t_{s-1}+1} \sum_{i=t_{s-1}}^{t_s} (y_i - \overline{y_s})^3}{S_s^3}, \tag{3}$$

where S_s is the standard deviation of the sth segment.

3. Kurtosis (γ_{2s}): This statistic is related to the degree of concentration that the values present around the mean of the distribution:

$$\gamma_{2s} = \frac{\frac{1}{t_s - t_{s-1}+1} \sum_{i=t_{s-1}}^{t_s} (y_i - \overline{y_s})^4}{S_s^4} - 3. \tag{4}$$

4. Slope of a linear regression of the points of the segment (a_s): A linear model is constructed for every segment trying to achieve the best linear approximation of the points of the time series in the evaluated segment. The slope is a measure of the general tendency of the segment:

$$a_s = \frac{S_s^{yt}}{\left(S_s^t\right)^2}, \tag{5}$$

where, for the sth segment, S_s^{yt} is the covariance between the time indexes, t, and the time series values, y; and S_s^t is the standard deviation of the time values. Covariance S_s^{yt} is defined by:

$$S_s^{yt} = \frac{1}{t_s - t_{s-1} + 1} \sum_{i=t_{s-1}}^{t_s} (i - \overline{t_s}) \cdot (y_i - \overline{y_s}). \tag{6}$$

5. Mean Squared Error (MSE_s): Considering the same linear model than the one used for the slope, we measure the error (MSE_s) of this linear fitting:

$$MSE_s = S_s^2 \cdot (1 - r_s^2), \text{ where } r_s^2 = \frac{S_s^{yt}}{S_s^2 \cdot \left(S_s^t\right)^2}. \tag{7}$$

6. Autocorrelation coefficient (AC_s): This a measure of the correlation between the current values of the time series and the previous ones:

$$AC_s = \frac{\sum_{i=t_{s-1}}^{t_s}(y_i - \overline{y_s}) \cdot (y_{i+1} - \overline{y_s})}{S_s^2}. \quad (8)$$

2.4.2. Clustering process: k*-means*

A clustering process is applied to group similar segments (taking into account the six selected statistical measures). For its simplicity and low computational load, the algorithm chosen for the clustering step is the well-known *k*-means. Before the clustering algorithm, note that a normalisation of the values of the segment metrics is conducted, as the distance from each segment to its centroid strongly depends on the range of values of each metric.

In the classic *k*-means, the initial centroids are randomly chosen from the set of patterns. Instead, we have developed a deterministic process to select these centroids which ensures that a chromosome will always have the same fitness. First, we choose the feature with the maximum standard deviation. The first initial centroid will be the segment with the highest value for this feature. The second one will be the segment with the highest Euclidean distance from the first centroid. The third centroid will be that which is farthest from both, and so on. This assures a deterministic initialisation, at the same time that the initial centroids are as far as possible from each other, favouring centroid diversity.

2.4.3. Measuring the quality of the clustering process

The last step of the evaluation of the chromosome is to measure how well the segments are grouped (compactness of the clustering). It is clear that different clustering algorithms usually lead to different clusters or reveal different clustering structures. In this sense, the problem of objectively and quantitatively evaluating the clustering results is particularly important, and this is known in the literature as cluster validation. There are two different testing criteria for this purpose [28,29]: external criteria and internal criteria. When a clustering is evaluated based on the clustered data it is called internal evaluation. In external evaluation, the clustering is evaluated using, for example, known class labels. Based on these concepts, the internal criteria evaluation metrics will be a suitable option for the evolution, because the GA is not given a priori information of the segments to be found. Note that the segments metrics are normalised at this step as well. We have considered ten different metrics:

- *Sum of squared errors (SSE)*: The simplest error measure is the sum of squared errors (considering errors as the distance from each point to their centroid). One of the disadvantages with the *SSE* measure is that it is sensitive to the number of segments in the solution (this index being generally lower as the number of segments decreases). Because of this, a more consistent approach would be to divide the error by the number of segments:

$$SSE = \frac{1}{m}\sum_{i=1}^{k}\sum_{\mathbf{x}\in\mathcal{C}_i} d(\mathbf{x}, \mathbf{c}_i)^2, \quad (9)$$

  where k is the number of clusters, $\mathbf{c}_i$ is the centroid of cluster $\mathcal{C}_i$, m is the number of segments and $d(\mathbf{x}, \mathbf{c}_i)$ is the Euclidean distance between pattern $\mathbf{x}$ and centroid $\mathbf{c}_i$. This function does not prevent clusters to fall very close in the clustering space. As this index has to be minimised, the fitness will be defined as $f = \frac{1}{1+SSE}$.
- *Normalised sum of squared errors (NSSE)*: Additionally, the distance between clusters should be maximised to promote well separated groups. This can be done by considering the following expression:

$$NSSE = \frac{\frac{1}{m}\sum_{i=1}^{k}\sum_{\mathbf{x}\in\mathcal{C}_i} d(\mathbf{x}, \mathbf{c}_i)^2}{(m-1)!\sum_{i=1}^{k}\sum_{j=i+1}^{k} d(\mathbf{c}_i, \mathbf{c}_j)}. \quad (10)$$

  The fitness in this case will be also defined as $f = \frac{1}{1+NSSE}$.
- *Caliński and Harabasz index (CH)*: This index has been found to be one of the best performing ones for adjusting the value of *k*. It is defined as:

$$CH = \frac{\mathtt{Tr}(\mathbf{S}_B) \cdot (N-k)}{\mathtt{Tr}(\mathbf{S}_W) \cdot (k-1)}, \quad (11)$$

  where N is the number of patterns, and $\mathtt{Tr}(\mathbf{S}_B)$ and $\mathtt{Tr}(\mathbf{S}_W)$ are the trace of the between and within-class scatter matrices, respectively. Note that the value of k will be fixed in our algorithm. As this index has to be maximised, the fitness will be defined as $f = CH$.
- *Davies–Bouldin index (DB)*: This index also attempts to maximize the between-cluster distance while minimising the distance from the cluster centroids to the rest of points. It is calculated as follows:

$$DB = \frac{1}{k}\sum_{i=1}^{k}\max_{i\neq j}\frac{\alpha_i + \alpha_j}{d(\mathbf{c}_i, \mathbf{c}_j)}, \quad (12)$$

  where α_i is the average distance of all elements in cluster $\mathcal{C}_i$ to centroid $\mathbf{c}_i$, and $d(\mathbf{c}_i, \mathbf{c}_j)$ is the distance between centroids $\mathbf{c}_i$ and $\mathbf{c}_j$. As this index has to be minimised, the fitness will be defined as $f = \frac{1}{1+DB}$.
- *Dunn index (DU) and variants*: The Dunn index attempts to identify clusters that are compact and well-separated. In this case, the distance between two clusters is defined as $\delta(\mathcal{C}_i, \mathcal{C}_j) = \min_{\mathbf{x}\in\mathcal{C}_i, \mathbf{y}\in\mathcal{C}_j} d(\mathbf{x}, \mathbf{y})$, that is, the minimum distance between a pair of points $\mathbf{x}$ and $\mathbf{y}$ belonging to $\mathcal{C}_i$ and $\mathcal{C}_j$. Furthermore, we define the diameter $\mathtt{diam}(\mathcal{C}_i)$ of cluster $\mathcal{C}_i$ as the maximum distance between two of its members, such as: $\mathtt{diam}(\mathcal{C}_i) = \max_{\mathbf{x},\mathbf{y}\in\mathcal{C}_i} d(\mathbf{x}, \mathbf{y})$. Then, the Dunn index is constructed as:

$$DU = \min_{i=1,\ldots,k}\left(\min_{j=i+1,\ldots,k}\left(\frac{\delta(\mathcal{C}_i, \mathcal{C}_j)}{\max_{l=1,\ldots,k}\mathtt{diam}(\mathcal{C}_l)}\right)\right). \quad (13)$$

  The Dunn index has been found to be sensitive to noise, but this disadvantage can be avoided by considering different definitions of cluster distance or cluster diameter. In this paper, we consider four different definitions of the Dunn index (corresponding to different definitions of the distance between clusters and the cluster diameter). For example, as suggested in [28], the cluster diameter can be computed as:

$$\mathtt{diam}(\mathcal{C}_i) = \frac{1}{N_{\mathcal{C}_i}(N_{\mathcal{C}_i}-1)}\sum_{\mathbf{x},\mathbf{y}\in\mathcal{C}_i} d(\mathbf{x}, \mathbf{y}), \quad (14)$$

  where $N_{\mathcal{C}_i}$ is the number of patterns belonging to cluster $\mathcal{C}_i$. This cluster diameter estimation has been found to be more robust in the presence of noise and as such, it has been included in the experiments (DU fitness function). Moreover, as said, other variants exist for this index. We selected the three variants that reported the best results in [29], referred to as GD33, GD43 and GD53, which correspond to the following variations of $\delta(\mathcal{C}_i, \mathcal{C}_j)$:

$$\delta^3(\mathcal{C}_i, \mathcal{C}_j) = \frac{1}{N_{\mathcal{C}_i}N_{\mathcal{C}_j}}\sum_{\mathbf{x}\in\mathcal{C}_i}\sum_{\mathbf{y}\in\mathcal{C}_j} d(\mathbf{x}, \mathbf{y}), \quad (15)$$

$$\delta^4(\mathcal{C}_i, \mathcal{C}_j) = d(\overline{\mathcal{C}}_i, \overline{\mathcal{C}}_j), \quad (16)$$

$$\delta^5(\mathcal{C}_i, \mathcal{C}_j) = \frac{1}{N_{\mathcal{C}_i} + N_{\mathcal{C}_j}}\left(\sum_{\mathbf{x}\in\mathcal{C}_i} d(\mathbf{x}, \overline{\mathcal{C}}_i) + \sum_{\mathbf{y}\in\mathcal{C}_j} d(\mathbf{y}, \overline{\mathcal{C}}_j)\right), \quad (17)$$

where $\overline{C}_i$ represents the mean of the cluster. The term $\texttt{diam}(C_i)$ is defined in this case as:

$$\texttt{diam}(C_i) = \frac{2}{N_{C_i}} \sum_{\mathbf{x} \in C_i} d^*_{ps}(\mathbf{x}, C_i), \tag{18}$$

where $d^*_{ps}(\mathbf{x}, C_i)$ is the Point Symmetry-Distance between object $\mathbf{x}$ and the cluster C_i defined in [30].
Since all the variations of this index have to be maximised, the fitness will be $f = DU$.

- *Silhouette index (SI)*: For this index, the data cohesion is computed based on the distance between all the points in the same cluster, and the separation is measured using the nearest neighbour distance. It is defined as:

$$SI = \frac{1}{N} \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \frac{b(\mathbf{x}, C_i) - a(\mathbf{x}, C_i)}{\max(a(\mathbf{x}, C_i), b(\mathbf{x}, C_i))}, \tag{19}$$

where a and b correspond, respectively, to the intra-cluster and inter-cluster distances and are defined as follows:

$$a(\mathbf{x}, C_i) = \frac{1}{N_{C_i}} \sum_{y \in C_i} d(\mathbf{x}, \mathbf{y}), \tag{20}$$

$$b(\mathbf{x}, C_i) = \min_{C_l, l \neq i} \left\{ \frac{1}{N_{C_l}} \sum_{\mathbf{y} \in C_l} d(\mathbf{x}, \mathbf{y}) \right\}. \tag{21}$$

- *COP index (COP)*: The cluster cohesion is estimated, for this index, using the distance from the points to their cluster centroid and the furthest neighbour distance. This index is defined as:

$$COP = \frac{1}{N} \sum_{i=1}^{k} \frac{\sum_{\mathbf{y} \in C_k} d(\mathbf{y}, \mathbf{c}_k)}{N_{C_k} \cdot \min_{\mathbf{x} \notin C_k} \max_{\mathbf{y} \in C_k} d(\mathbf{x}, \mathbf{y})}. \tag{22}$$

2.5. Selection and replacement processes

All individuals will be considered for reproduction and generation of offspring, promoting a greater diversity because all individuals are potential parents. After the application of the genetic operators, the offspring and the parent population are joined and a replacement process is performed by roulette wheel selection. The selection probability for each individual chromosome is calculated from its fitness value. The roulette wheel process is repeated as many times as the population size minus one, and the last place is kept for the best segmentation of the previous generation, thus being an elitist algorithm. As can be seen, the selection process promotes diversity, while the replacement process promotes elitism.

2.6. Crossover operator

The algorithm includes a crossover operator (represented in Fig. 2), whose main function is to perform an exploitation of the existing solutions. For each parent individual, the crossover operator is applied with a given probability p_c. The operator randomly selects a different parent and a time index. Then, the left and right parts of the selected chromosomes are interchanged with respect to the time index. After this step, the algorithm checks if the solutions verify the minimum length per segment constraint. If not, the crossover operator is repeated using the same parents, until a maximum of three attempts is reached. If no feasible solution is produced, a different parent individual is chosen. This process is repeated until the crossover ends successfully.

2.7. Mutation operator

Two mutation operators are included in the GA with the aim of reducing the dependency on the initial population and for escaping from local optima. The probability p_m of performing any mutation is decided by the user. When a mutation is to be performed, the type of perturbation is randomly selected from the following two: (1) add or remove (with the same probability) a given number of cut points; and (2) move a given number of cut points towards the left or the right (with the same probability).

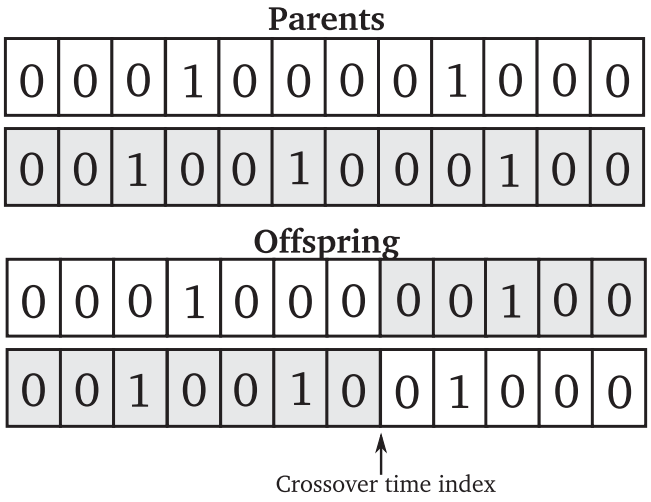


Fig. 2. Representation of the crossover operator: a time index is randomly chosen as cut point, and the resulting offspring is generated.

For all the mutations, the number of cut points to be mutated is decided by an user parameter (m_p) as a percentage of the current number of cut points. When moving cut points to the right or to the left, each selected cut point is randomly pushed towards the previous or the following cut point (with the constraint that it has to maintain the minimum segment length). If the mutation does not maintain this constraint, the same procedure used in the crossover is considered.

2.8. Parameters of the algorithm

For the sake of clarity, this section includes a summary of all the parameters involved in the proposed segmentation algorithm. Concerning the clustering step, the two parameters are k (the number of clusters to be found) and *it* (number of iterations for k-means). With respect to the evolutionary aspect of the algorithm, we have a wider range of parameters: g (number of generations), P (population size), p_m (mutation probability), p_c (crossover probability), s (seed), m_p (percentage of cut points to be mutated), $\overline{sl}$ (initial mean segment length) and f (fitness function).

3. Automatic evaluation of the segmentation

In order to evaluate the results of the algorithm, two evaluation metrics are proposed. These measures analyse both the homogeneity of cluster assignation with respect to the DO events and the robustness of the results obtained from different seeds. They are not included in the fitness function, serving only as an automatic way of evaluating the quality of the segmentation, avoiding the intervention of the expert. Both indexes compare two different clustering partitions:

1. *Rand index (RI)*: This metric is particularly useful for data clustering evaluation [31]. It is related to the accuracy or correct classification rate (since it measures the correlation between the obtained clusters and the ideal ones), but it is applicable even when the exact class labels are not available, as in our case. A set $Y = \{y_n\}_{n=1}^{N}$ is given (in our case, the time series), and two clustering partitions of Y are to be compared:

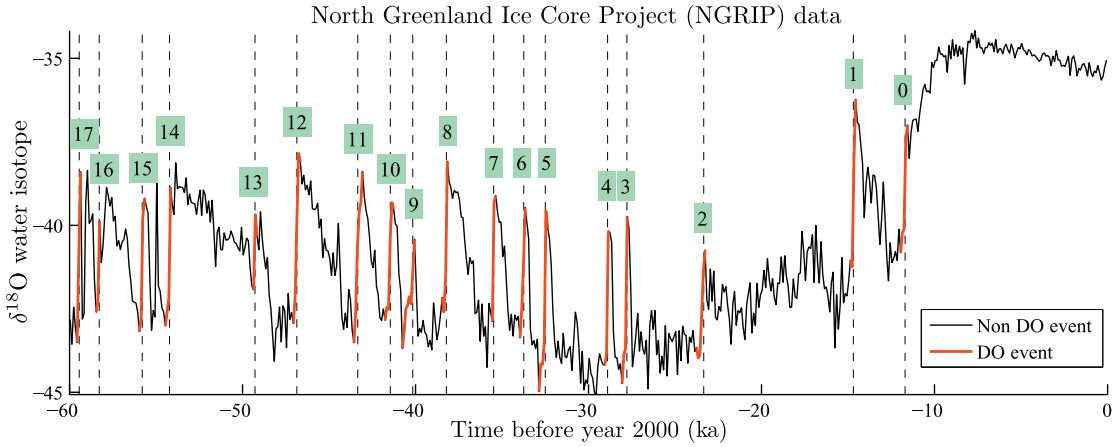


Fig. 3. Representation of the ideal segmentation and the different DO events for the North Greenland Ice Core Project (NGRIP).

$X = \{X_1, \ldots, X_r\}$ and $Z = \{Z_1, \ldots, Z_s\}$. For a given segmentation, the partitions are defined in the following way: X_l is a set containing every $y_i \in s_s, s_s \in \mathcal{C}_l$, i.e. the partitions are based on the label assigned to each time series value y_i from the current segmentation. The following two numbers are defined: a (number of pairs in Y that are in the same set in X and Z) and b (number of pairs in Y that are in different sets in X and Z). Then, the Rand index is defined as: $RI = (a+b)/\binom{N}{2}$. This metric has a value between 0 and 1, with 0 indicating that the two partitions do not agree on any pair of points and 1 indicating that they are exactly the same.

2. *Adjusted rand index (ARI)*: It is a corrected version of the RI [32] trying to fix some of its known problems, e.g. the expected value of the RI for two random partitions does not take a constant value and it approaches its upper limit of unity as the number of clusters increases. ARI values range from -1 to $+1$, yielding negative values if the index is less than the expected one. The detailed formulation can be found in [32].

In order to evaluate the segmentation returned by the algorithm, we compare it with an ideal segmentation.[3] These ideal segmentations have been designed by examining the literature about Dansgaard–Oeschger (DO) events, which are associated to TPs. The ideal segmentation for NGRIP has been included in Fig. 3. In the figures, the onsets of the DO events (in a first approximation, we do not consider the error margin) reported in [23] are represented by vertical lines and the segments covering the precursor period to the DO events are delimited by the slope closest to the corresponding onset. The closer the segmentation returned by the GA is to this ideal segmentation, the better the segmentation. To perform this comparison, RI and ARI indexes will be used (ARI_Ideal and RI_Ideal).

Given that the wishful ideal segmentation would be binary (non DO event or DO event) and the segmentation returned by the GA can have a value of $k > 2$, we need to binarise the segmentation of the GA (i.e. decide which clusters represent the DO events and which not). Preliminary experiments [17] revealed that DO events were usually identified by two different sequences of clusters, so we evaluated ARI_Ideal and RI_Ideal for all possible combinations of one or two clusters. The final value was the maximum ARI_Ideal and RI_Ideal values of all these combinations.

[3] It is a hypothetically ideal segmentation, based on the available data. The hypothesis is that the onset of the DO events is detected from combined analysis of benthic sediment and ice core data [33]. Those data do not always agree, which could result in part of the error margin. The timing method contributes the rest of the error.

Additionally, the stability of the GA was estimated by comparing the 30 segmentations from the 30 different runs. This was done by averaging RI and ARI comparing all possible pairs of segmentations (ARI_Seeds and RI_Seeds).

4. Experiments

As stated before, we study and analyse Dansgaard–Oeschger (DO) events (i.e. the result of different tipping points occurred in the past). To do so, the paleoclimate datasets chosen for this study are the Greenland Ice Sheet Project Two (GISP2) and the North Greenland Ice Core Project (NGRIP) δ^{18}O ice core data [22,23]. The δ^{18}O water isotope record is used as a proxy for past atmospheric temperature variations. We focus on the 20-yr resolution δ^{18}O isotope records from both drilling sites. The dataset is pre-processed by computing a 5-point average to reduce short-term fluctuations within the data. In this way, the time series considered is $\{y^*_n\}_{n=1}^{N/5}$ with $y^*_i = \frac{1}{5}\sum_{j=5i}^{5i+4} y_i$.

This paper considers three different objectives, which are explored in this section: (1) to test our segmentation algorithm which provides a set of clustered time segments; (2) to analyse the result of this clustering studying the clusters that best represent TPs; and, finally, (3) to study whether future TPs can be predicted via a classification approach.

The source code in Matlab for the methodology developed in this paper will be made available upon acceptance of the manuscript.

4.1. Experimental setting

It is well-known that GAs usually involve adjusting a notable set of parameters. However, their search dynamics can easily adapt to different problems, resulting in a performance that is negligibly affected by minor changes in the parameters. The GA was configured with the following parameters: the number of individuals of the population is $P = 100$. The crossover probability is $p_c = 0.8$ and the mutation probability $p_m = 0.2$. The percentage of cut points to be mutated (m_p) is the integer part of the 20% of the number of cut points, and the average segment length for the initialisation is $\overline{sl} = 4$. The maximum number of generations is set to $g = 100$, and the k-means clustering process is allowed a maximum of 20 iterations (it). These parameters were optimised by a trial and error procedure, although the algorithm showed very robust performance to their values. The most important parameters for the final performance of the algorithm were $\overline{sl}$ and k.

We performed different experiments considering the 10 different fitness functions presented in Section 2.4.3. The value of $k = 5$

Table 1
Mean and standard deviation segmentation results for the different clustering indexes and time series used.

Fitness	ARI_Ideal	RI_Ideal	ARI_Seeds	RI_Seeds
NGRIP time series				
DB	0.330 ± 0.079	0.791 ± 0.021	*0.368 ± 0.079*	0.729 ± 0.039
DU	0.307 ± 0.056	0.793 ± 0.014	$\mathbf{0.381 \pm 0.090}$	$\mathbf{0.737 \pm 0.042}$
CH	$\mathbf{0.429 \pm 0.074}$	$\mathbf{0.823 \pm 0.019}$	0.110 ± 0.031	0.602 ± 0.019
SSE	0.349 ± 0.068	0.793 ± 0.023	0.130 ± 0.033	0.613 ± 0.015
NSSE	0.332 ± 0.072	0.795 ± 0.017	0.126 ± 0.030	0.593 ± 0.023
SH	0.270 ± 0.050	0.787 ± 0.010	*0.368 ± 0.095*	0.712 ± 0.045
GD33	0.288 ± 0.062	0.791 ± 0.012	0.357 ± 0.091	*0.731 ± 0.044*
GD43	0.289 ± 0.059	0.792 ± 0.012	0.356 ± 0.091	0.726 ± 0.045
GD53	0.339 ± 0.063	0.797 ± 0.014	0.149 ± 0.070	0.573 ± 0.047
COP	*0.394 ± 0.052*	*0.811 ± 0.020*	0.103 ± 0.035	0.579 ± 0.024
GISP2 time series				
DB	0.365 ± 0.073	0.788 ± 0.023	*0.363 ± 0.082*	*0.716 ± 0.039*
DU	0.363 ± 0.069	0.794 ± 0.016	0.360 ± 0.087	$\mathbf{0.718 \pm 0.042}$
CH	$\mathbf{0.448 \pm 0.063}$	$\mathbf{0.817 \pm 0.023}$	0.121 ± 0.028	0.601 ± 0.038
SSE	0.348 ± 0.055	0.788 ± 0.017	0.119 ± 0.030	0.620 ± 0.015
NSSE	0.354 ± 0.066	0.775 ± 0.025	0.133 ± 0.034	0.601 ± 0.025
SH	0.296 ± 0.051	0.775 ± 0.012	$\mathbf{0.374 \pm 0.087}$	0.702 ± 0.044
GD33	0.331 ± 0.068	0.792 ± 0.018	0.303 ± 0.081	0.689 ± 0.041
GD43	0.325 ± 0.079	0.792 ± 0.017	0.311 ± 0.096	0.698 ± 0.046
GD53	0.377 ± 0.073	*0.797 ± 0.019*	0.148 ± 0.066	0.577 ± 0.038
COP	*0.385 ± 0.067*	0.796 ± 0.021	0.104 ± 0.031	0.576 ± 0.032

for the k-means algorithm has been chosen given the previous good results in the NGRIP data [34]. This result indicates that the concept and nature of DO events is too complex to only consider a binary approach. The climate system exhibits a dynamical behaviour with intrinsic variability hence a binary approach is not able to encompass all features present within a DO event, being $k = 5$ a reasonable choice. Moreover, the method can group several DO events together and is still a useful tool to better understand the behaviour of DO events. It is important to recall that the algorithm estimates the optimal segments without any prior information of the DO events. The only information given to the algorithm is the time series and the statistics to use for the clustering (to validate whether the statistics proposed in the literature are useful for characterising paleoclimate TPs and DO events in general). The best segmentation returned by the GA in the last generation is analysed by an evaluation metric based on the onsets of the DO events. This evaluation metric will be presented in the next section.

Finally, since GAs are stochastic optimisation algorithms with a embedded random number generator, their results can be different depending on the seed. To obtain more robust results, the algorithm is run several times with different seeds. For each dataset, the GA was run 30 times with different seeds.

4.2. Segmentation results

The results of these experiments for the two datasets are included in Table 1. As can be seen, the fitness function that results in the best set of segmentations for both datasets is the *CH* index, which achieves very promising performance. However, as indicated by the ARI_Seeds and RI_Seeds metrics, it presents a low stability. In this sense, both *DB* and *DU* present an acceptable trade-off between all evaluation metrics.

To properly analyse the evolution process of our algorithm, we have compared the initial and final solutions using the seed that provides the best results with the *CH* fitness function, resulting in the following performance: For NGRIP the initial solution obtained 0.371 and 0.812 for ARI_Ideal and RI_Ideal, respectively, and the final solution obtained 0.565 and 0.855. For GISP2, the initial solution resulted in a 0.394 of ARI_Ideal and a 0.781 of RI_Ideal, while the final solution obtained 0.576 and 0.864. This performance demonstrates the capability of the proposed algorithm to capture statistical features of different time segments and cluster together similar patterns, specially in relation to paleoclimate data.

The segmentations obtaining the highest ARI_Ideal metric for the *CH* fitness function (with a value of 0.565 and 0.576 for NGRIP and GISP2, respectively), along with a representation of the 18 DO events can be seen in Figs. 4 and 5. The segments have been coloured according to their cluster assignment. Moreover, Figs. 6 and 7 include the individual cut points obtained by the algorithm. Note that those segments belonging to the same cluster in Figs. 6 and 7 are grouped together in Figs. 4 and 5. For the sake of visualisation and to better appreciate the degree of detail of the segmentations obtained by the algorithm, Fig. 8 includes a zoom of Fig. 6. The clusters related to the DO events are $\mathcal{C}_2$ and $\mathcal{C}_4$ in both time series. If one compares the NGRIP segmentation to the ideal one in Fig. 3, it can be seen that almost all DO events are correctly segmented by the algorithm ($\mathcal{C}_2$ and $\mathcal{C}_4$ segments are usually close to the DO onset) and that there are very few "false positives" labels ($\mathcal{C}_2$ and $\mathcal{C}_4$ segments are not found in a non DO event part of the series). For the case of NGRIP, there are four events that are not detected: 2, 9, 13 and 16. In the case of GISP2 there are only two events that have not been detected: 9 and 13.

The clustering space of the best segmentations produced can be analysed in Figs. 9 and 10. These figures confirm that there are obvious differences between the two clusters associated to the DO events ($\mathcal{C}_2$ and $\mathcal{C}_4$), mainly from the values of the MSE_s and a_s metrics.

4.3. Prediction experiment

As stated before, the main motivation for using a segmentation algorithm is to produce a set of time segments according to a specific criteria (in our case a set of statistical features). This segmentation will provide the expert with a predefined set of time spans to analyse and study different patterns or events (e.g. an analysis of the time segments associated to TPs can be used to validate the hypotheses in the literature about the increase in the different statistics). In relation to paleoclimate data, tipping points have been studied using different time-windows (usually, each window is fixed depending on the DO event). In this section, we validate the hypothesis that the time segments can be helpful both for understanding the problem considered and for constructing a diagnostic–predictive model (which could anticipate a new TP).

For this experiment the time segments produced by the GA have been binarised considering whether they represent a DO event. Note that the first segment was not included in the analysis (as there was no information available about previous points). The main objective in this case is to construct an interpretable classification model for this binary variable (to complement the literature about the most influential statistics to detect TPs). More specifically, the classification method will consider the information preceding a given time segment s_s and try to predict if this time segment s_s is associated to a DO event. In the analysis of the results of this paper, it has been seen that not all TPs are classified in the same cluster or clusters (and, therefore, they present different statistical values). However, there are clusters that are mainly composed of TPs, meaning this that there are indeed a series of characteristics that define some of the TPs considered, but the rest of TPs do not follow this characterisation, which could mean that they can be considered as outliers or that the statistics needed to characterise them have not been discovered yet. Because of this, it could be interesting to use as an additional feature for the classification of the cluster associated to the previous segment. Thus, we propose to use different sources of information for detecting TPs: the cluster (label) of the previous segment, the statistics of the previous segment and a set of previous points. Given that the

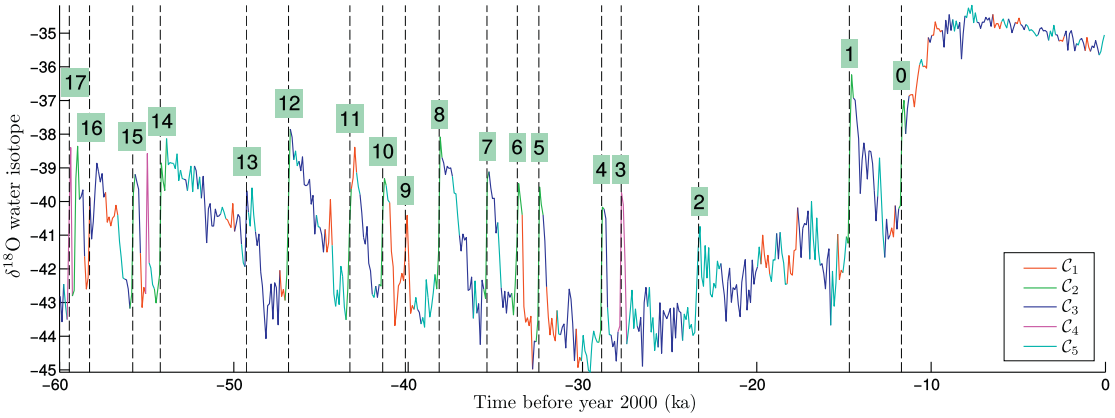


Fig. 4. Best cluster assignation obtained in the NGRIP dataset after the evolutionary process using the *CH* clustering index.

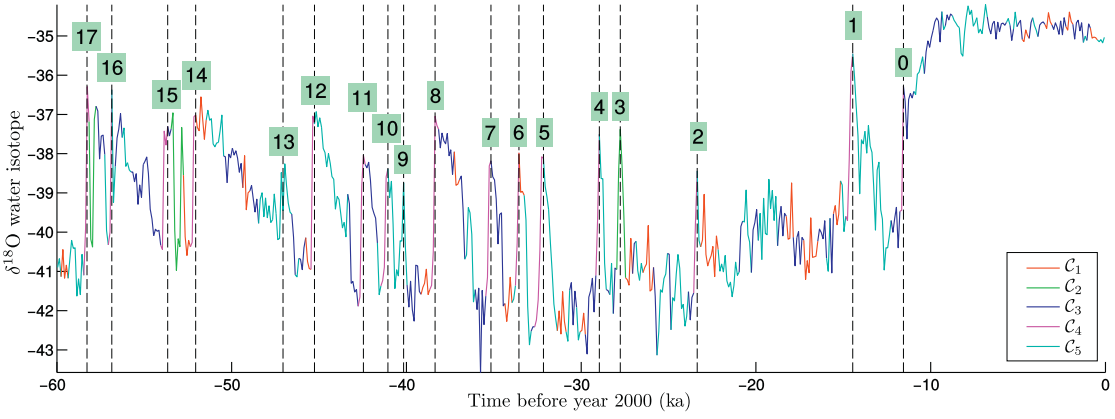


Fig. 5. Best cluster assignation obtained in the GISP2 dataset after the evolutionary process using the *CH* clustering index.

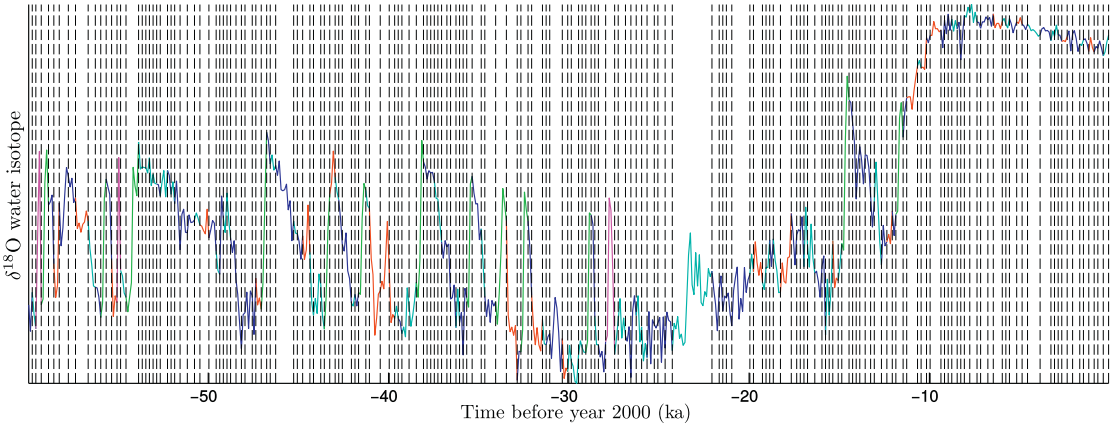


Fig. 6. Time cuts obtained for the best cluster assignation obtained in the NGRIP dataset.

onset of the DO events lasts several hundreds of years, as can be seen in the time series, the segment under evaluation should have a length of at least the same order of magnitude, in order to enable the diagnosis and prediction of its class.

As stated before, different input features can be considered as arguments for the diagnostic–prediction step. Firstly, the most direct and widely used approach is to consider a fixed time-window. Secondly, a dynamic-window can be considered (where the window is the size of each segment and the features used are the associated statistics and cluster label). Our proposal is to mix both sources of information (not only use the statistical features associated to the previous segment, but also a fixed time-window, which will ideally complement the statistics with finer-grain information). In this sense, the "Fixed-window" approach only uses as features for the classification the points previous to the given segment; the "Dynamic-window" uses the statistics of the previous segment (independently, of its size) and the associated cluster label; our approach considers these three sources of information.

The results have been reported in terms of two metrics, one of them specially designed to deal with imbalanced datasets:

1. The well-known Accuracy metric (*Acc*), that corresponds to the ratio of correctly classified patterns and measures overall performance.

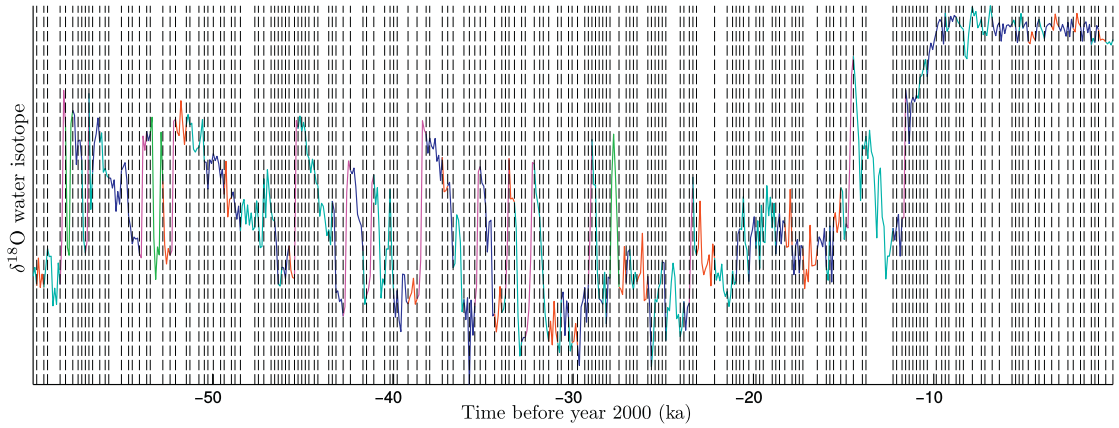


Fig. 7. Time cuts obtained for the best cluster assignation obtained in the GISP2 dataset.

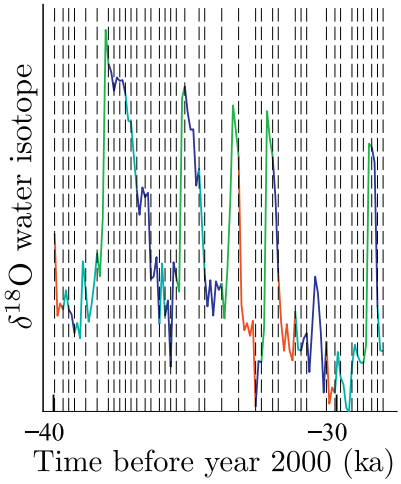


Fig. 8. Zoomed representation of the best clustering obtained in the NGRIP series after the evolutionary process using the *CH* clustering index.

2. The Geometric Mean of the sensitivities ($GM = 100 \cdot \sqrt{S_p \cdot S_n}$), where S_p is the sensitivity for the positive class (ratio of correctly classified patterns considering only this class) and S_n is the sensitivity for the negative one. The importance of this metric lies in the imbalanced nature of the dataset (note that we have information related to only 18 DO events).

The algorithm considered is a C4.5 decision tree, in order to interpret the results and provide domain-knowledge. The confidence factor used for pruning is set to 0.25 and the minimum number of objects per leaf to 1. The results associated to this experiment can be seen in Table 2, where the results are provided using all the data both for training and testing the model (this experiment is referred to as Train in the table) and using a leave-one-out approach (LOO column in the table). With the purpose of extracting domain-knowledge we consider that the Train approach describes better the training data (possibly incurring in overfitting), while the LOO approach can also provide information about the generalisation of the model. The "Fixed-window" approach in Table 2 considers the three time series values preceding each segment (i.e. the prediction is performed using only three features). The "Dynamic-window" approach includes the six statistics and the cluster assignment of the segment preceding the one to be predicted (i.e. seven features). The "Combination" approach combines both sources of information, thus resulting in a set of 10 features. In the case of the fixed-window approach, different window sizes

Table 2
Acc and *GM* results for the prediction of DO events using the NGRIP and GISP2 time series.

NGRIP	Train		LOO	
Features	*Acc*	*GM*	*Acc*	*GM*
Fixed-window	90.16	0.00	**90.16**	0.00
Dynamic-window	90.16	0.00	89.07	0.00
Combination	**93.99**	**62.26**	89.62	**40.81**
GISP2	Train		LOO	
Features	*Acc*	*GM*	*Acc*	*GM*
Fixed-window	89.94	0.00	89.94	0.00
Dynamic-window	94.97	70.71	86.59	0.00
Combination	**100.00**	**100.00**	**91.06**	**69.28**

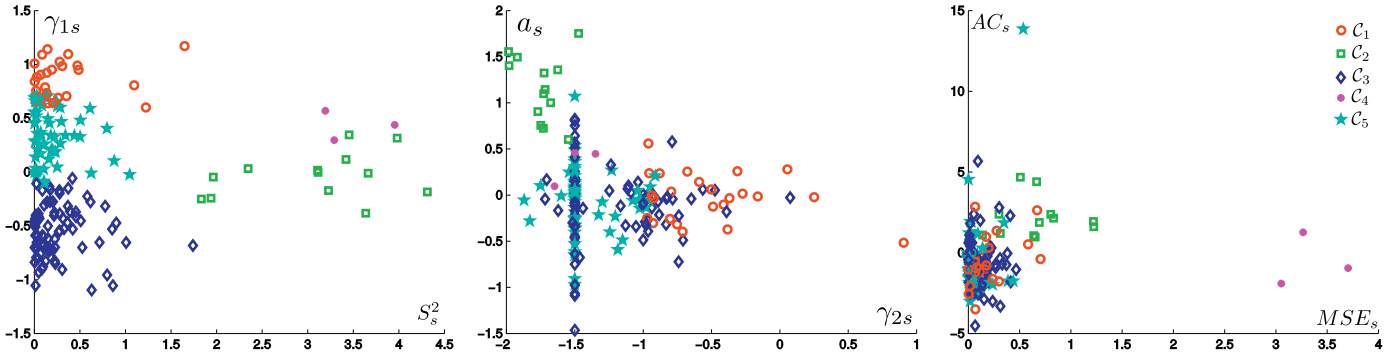


Fig. 9. NGRIP clustering space for the six statistical metrics considered (each point represents a segment).

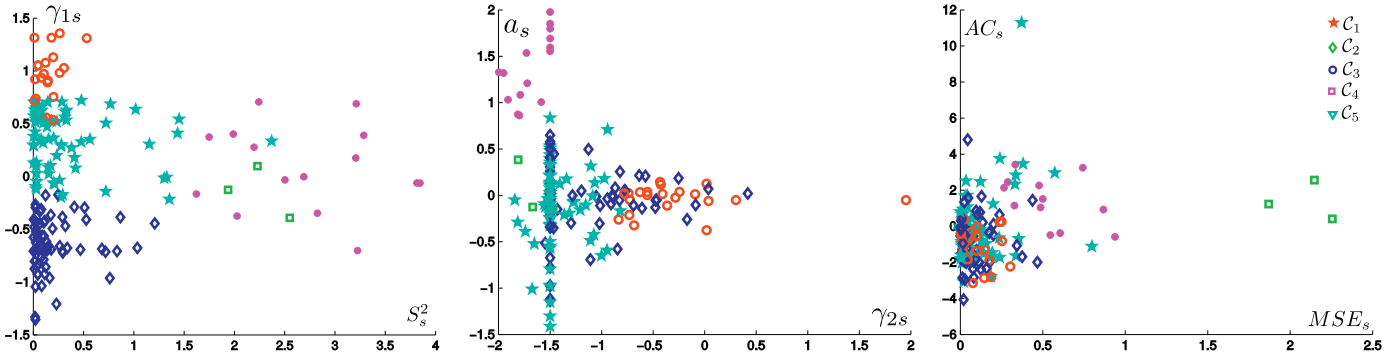


Fig. 10. GISP2 clustering space for the six statistical metrics considered (each point represents a segment).

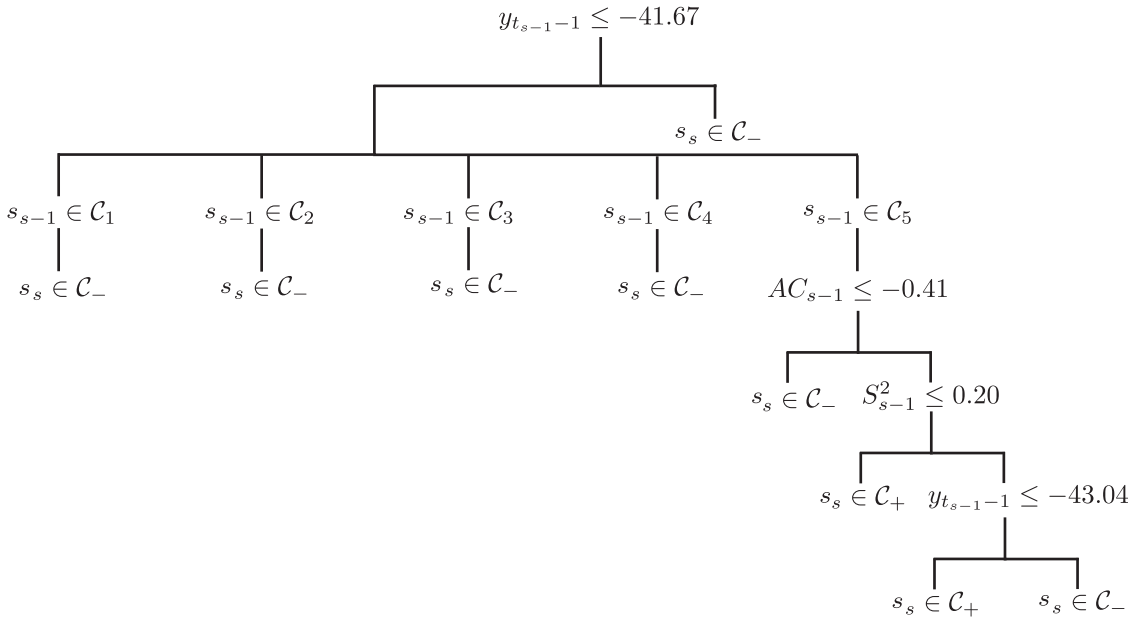


Fig. 11. Representation of the decision tree obtained in the complete dataset for the NGRIP time series.

were tested, although the results were the same in all the cases. For our proposal, we checked that increasing the size of the window did not have an influence in the model.

Several conclusions can be drawn from this experiment. Firstly, it can be noted that in both cases the fixed-window approach provides a trivial solution (resulting in a model that predicts that all time segments are non DO events). In other words, none of the DO events are classified (which can be inferred by the *GM* result). This confirms the previous claim in the literature that all TPs cannot be predicted with the same window size and that the proposed statistics can be used as triggers for TPs. Concerning the sole use of the statistics (i.e. the "Dynamic-window" approach), the result in GISP2 for Train is more promising, although, as seen, it does not generalise well on unseen data, which could mean that these statistics can be a good predictor for past TPs, but they lack generalisation capability. However, the combination of both sources of information (values of the time series using a fixed-window and statistics computed over a dynamic-window) is successful (specially for GISP2, which was the time series in which the GA obtained the best segmentation results). Another important result is the fact that the segmentation algorithm results in time segments and clusters that are helpful for the prediction and that a diagnostic–predictive model is feasible in the context considered.

The decision trees obtained from this experiment can be seen in Figs. 11 and 12. Each node of the tree represents a condition. When the condition is met, the right part of the tree is chosen. The two classes considered are: the negative class or $\mathcal{C}_-$ (associated to non DO events) and the positive class or $\mathcal{C}_+$ (DO events). Note that for the prediction of the class associated to a segment s_s the characteristics considered are those associated to segment s_{s-1} and the three time instants before s_s (i.e., $y_{t_{s-1}-1}, y_{t_{s-1}-2}, y_{t_{s-1}-3}$). Note that the conditions in the decision tree are relative comparisons between the existing classes.

For both trees, the time instant previous to the beginning of the segment ($y_{t_{s-1}-1}$) is the most discriminating feature (as it is the one that appears at the top of the tree) and is also used in other nodes of the trees. In the case of NGRIP (Fig. 11), the cluster of the previous segment is an important characteristic, being DOs usually preceded by $\mathcal{C}_5$. The used statistics are in this case the autocorrelation and the variance. For GISP2 (Fig. 12) the used statistics are kurtosis, skewness and slope. There are two possible outcomes for segment s_s, it either belongs already to a DO onset segment which is found by comparison to the statistics encountered in the previous segment, or it does not belong to the DO onset, something that could change when the next observation in time is available (i.e. the length of segment s_{s-1} increases by one additional point). That could change significantly the statistical properties of the extended

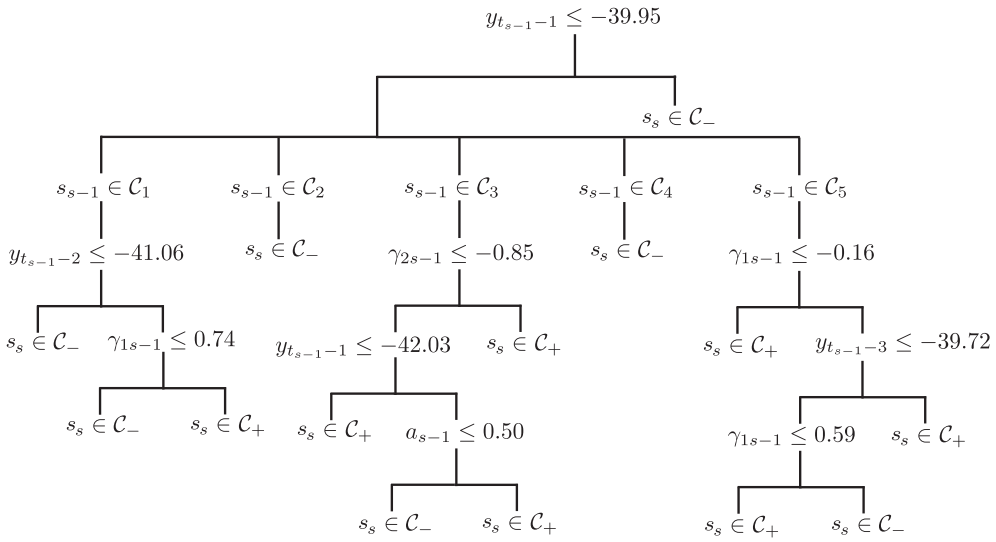


Fig. 12. Representation of the decision tree obtained in the complete dataset for the GISP2 time series.

segment. At the addition of any point in observation the segment evaluation should be repeated. It is important to emphasize once again the duration of a segment before being characterised to one of the classes of the DO sequence that exceeds the single next time step horizon.

5. Conclusions

This work tackles the problem of time series segmentation in the context of paleoclimate time series analysis. We propose a Genetic Algorithm for time series segmentation. which also clusters the time series by using six statistic characteristics that have been found to reveal incoming tipping points (TPs). As opposed to other proposals in the literature, which design TP-specific descriptors, our algorithm considers a generic approach (which has been shown to detect most abrupt climate changes in the series considered). Our experiments also demonstrate that the development of a machine learning-based diagnostic–predictive model is feasible and leads to very promising results given that the predictive-monitoring period of the system is several decades long and that the time series contains already abrupt climate changes of the same type in its record. The above aspects open an interesting avenue for future research.

Future work includes extending the segmentation method to consider simultaneously other related time series to create a more robust and generic decision tree in order to extract domain-knowledge. Finally, other clustering strategies can be helpful for the algorithm, e.g. the DBSCAN algorithm, which provides an outlier detection framework (which could be helpful in a time series as the one considered where there could be noise and very different influencing factors).

Acknowledgements

This work has been partially subsidized by the TIN2014-54583-C2-1-R project of the Spanish Ministerial Commission of Science and Technology (MINECO), FEDER funds and the P2011-TIC-7508 project of the “Junta de Andalucía” (Spain). A. Durán-Rosal’s research has been subsidized by the FPU Predoctoral Program (Spanish Ministry of Education and Science), grant reference FPU14/03039. The authors M. Pérez-Ortiz and A.M. Durán-Rosal have contributed equally to the preparation of this paper. This work has been performed within the Ariadna project 13-9202 of the European Space Agency.

References

[1] P. Wassmann, T. Lenton, Arctic tipping points in an earth system perspective, AMBIO 41 (1) (2012) 1–9.
[2] M. Allen, Planetary boundaries: tangible targets are critical, Nat. Rep. Clim. Chang. 3 (2009) 114–115.
[3] L. Carro-Calvo, S. Salcedo-Sanz, J. Luterbacher, Neural computation in paleoclimatology: general methodology and a case study, Neurocomputing 113 (2013) 262–268.
[4] T.M. Lenton, Early warning of climate tipping points, Nat. Clim. Chang. 1 (4) (2011) 201–209.
[5] V. Dakos, S.R. Carpenter, W.A. Brock, A.M. Ellison, V. Guttal, A.R. Ives, S. Kefi, V. Livina, D.A. Seekell, E.H. Van Nes, et al., Methods for detecting early warnings of critical transitions in time series illustrated using simulated ecological data, PLoS One 7 (7) (2012) e41010.
[6] Y. Lu, Z. Ju, Y. Liu, Y. Shen, H. Liu, Time series modeling of surface EMG based hand manipulation identification via expectation maximization algorithm, Neurocomputing 168 (2015) 661–668, doi:10.1016/j.neucom.2015.05.058.
[7] E.J. Keogh, S. Chu, D. Hart, M. Pazzani, Segmenting time series: a survey and novel approach, in: M. Last, A. Kandel, H. Bunke (Eds.), Data Mining In Time Series Databases, Series in Machine Perception and Artificial Intelligence, 57, World Scientific Publishing Company, 2004, pp. 1–22.
[8] E. Keogh, S. Chu, D. Hart, M. Pazzani, An online algorithm for segmenting time series, in: Proceedings of the IEEE International Conference on Data Mining (ICDM 2001), 2001, pp. 289–296.
[9] T.-c Fu, A review on time series data mining, Eng. Appl. Artif. Intell. 24 (1) (2011) 164–181.
[10] I. Khan, J.Z. Huang, K. Ivanov, Incremental density-based ensemble clustering over evolving data streams, Neurocomputing 191 (2016) 34–43.
[11] A.P. Serra, L.E. Zárate, Characterization of time series for analyzing of the evolution of time series clusters, Expert Syst. Appl. 42 (1) (2015) 596–611.
[12] E. Fuchs, T. Gruber, J. Nitschke, B. Sick, On-line motif detection in time series with SwiftMotif, Pattern Recognit. 42 (11) (2009) 3015–3031.
[13] H. Zhao, Z. Dong, T. Li, X. Wang, C. Pang, Segmenting time series with connected lines under maximum error bound, Inf. Sci. 345 (2016) 1–8.
[14] A.M. Durán-Rosal, P.A. Gutiérrez-Peña, F.J. Martínez-Estudillo, C. Hervás-Martínez, in: Hybrid artificial intelligent systems: 11th international conference (HAIS), Springer International Publishing, 2016, pp. 163–173.
[15] K.D. Bennett, Determination of the number of zones in a biostratigraphical sequence, New Phytol. 132 (1) (1996) 155–170.
[16] P. Prandom, M. Goodwin, M. Vetterli, Optimal time segmentation for signal modeling and compression, in: Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-97), 3, IEEE, 1997, pp. 2029–2032.
[17] A. Nikolaou, P. Gutiérrez, A. Durán, I. Dicaire, F. Fernández-Navarro, C. Hervás-Martínez, Detection of early warning signals in paleoclimate data using a genetic time series segmentation algorithm, Clim. Dyn. 44 (7–8) (2015) 1919–1933.
[18] R. Palivonaite, K. Lukoseviciute, M. Ragulskis, Short-term time series algebraic forecasting with mixed smoothing, Neurocomputing 171 (2016) 854–865.
[19] R. Palivonaite, M. Ragulskis, Short-term time series algebraic forecasting with internal smoothing, Neurocomputing 127 (2014) 161–171.
[20] R. Palivonaite, K. Lukoseviciute, M. Ragulskis, Algebraic segmentation of short nonstationary time series based on evolutionary prediction algorithms, Neurocomputing 121 (2013) 354–364.

[21] S. Aghabozorgi, A.S. Shirkhorshidi, T.Y. Wah, Time-series clustering a decade review, Inf. Syst. 53 (2015) 16–38.
[22] K.K. Andersen, N. Azuma, J.-M. Barnola, M. Bigler, P. Biscaye, N. Caillon, J. Chappellaz, H.B. Clausen, D. Dahl-Jensen, H. Fischer, et al., High-resolution record of northern hemisphere climate extending into the last interglacial period, Nature 431 (7005) (2004) 147–151.
[23] A. Svensson, K.K. Andersen, M. Bigler, H.B. Clausen, D. Dahl-Jensen, S.M. Davies, S.J. Johnsen, R. Muscheler, F. Parrenin, S.O. Rasmussen, R. Röthlisberger, I. Seierstad, J.P. Steffensen, B.M. Vinther, A 60,000 year greenland stratigraphic ice core chronology, Clim. Past 4 (1) (2008) 47–57.
[24] V.S. Tseng, C.-H. Chen, P.-C. Huang, T.-P. Hong, Cluster-based genetic segmentation of time series with DWT, Pattern Recognit. Lett. 30 (13) (2009) 1190–1197.
[25] S.L. Sclove, Time-series segmentation: a model and a method, Inf. Sci. 29 (1) (1983) 7–25.
[26] J. Himberg, K. Korpiaho, H. Mannila, J. Tikanmaki, H.T. Toivonen, Time series segmentation for context recognition in mobile devices, in: Proceedings of the IEEE International Conference on Data Mining (ICDM 2001), 2001, pp. 203–210.
[27] F.-L. Chung, T.-C. Fu, V. Ng, R.W. Luk, An evolutionary approach to pattern-based time series segmentation, IEEE Trans. Evol. Comput. 8 (5) (2004) 471–489.
[28] R. Xu, D. Wunsch, Clustering, IEEE Press Series on Computational Intelligence, Wiley, 2008.
[29] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J.M. Perez, I.n. Perona, An extensive comparative study of cluster validity indices, Pattern Recognit. 46 (1) (2013) 243–256.
[30] S. Bandyopadhyay, S. Saha, A point symmetry-based clustering technique for automatic evolution of clusters, IEEE Trans. Knowl. Data Eng. 20 (11) (2008) 1441–1457.
[31] W.M. Rand, Objective criteria for the evaluation of clustering methods, J. Am. Stat. Assoc. 66 (336) (1971) 846–850.
[32] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1) (1985) 193–218.
[33] L.C. Peterson, G.H. Haug, K.A. Hughen, U. Röhl, Rapid changes in the hydrologic cycle of the tropical atlantic during the last glacial, Science 290 (5498) (2000) 1947–1951.
[34] M. Pérez-Ortiz, P. Gutiérrez, J. Sánchez-Monedero, C. Hervás-Martínez, A. Nikolaou, I. Dicaire, F. Fernández-Navarro, Time series segmentation of paleoclimate tipping points by an evolutionary algorithm, in: Hybrid Artificial Intelligence Systems, in: Lecture Notes in Computer Science, 8480, Springer International Publishing, 2014, pp. 318–329.

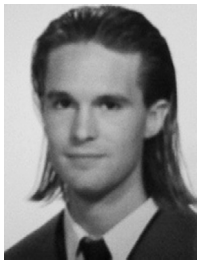
María Pérez-Ortiz was born in Cordoba, Spain. She received the B.S. degree in Computer Science in 2011 and the M.Sc. degree in Intelligent Systems in 2012 from the University of Córdoba, Spain. In 2015, she obtained the Ph.D. degree in Computer Science and Artificial Intelligence in the University of Córdoba. She is working as a researcher with the Department of Mathematics and Engineering (for Department of Quantitative Methods) at the Universidad Loyola Andalucía. Her current interests include a wide range of topics concerning machine learning and pattern recognition.

Antonio M. Durán-Rosal was born in Córdoba, Spain. He received the B.S. degree in Computer Science in 2014 and the M.Sc. degree in Computer Science in 2015 from the University of Córdoba, Spain, where he is currently pursuing the Ph.D. degree in Computer Science and Artificial Intelligence in the Department of Computer Science and Numerical Analysis. His current interests include a wide range of topics concerning machine learning, pattern recognition and time series analysis.

Pedro Antonio Gutiérrez was born in Cordoba, Spain. He received the B.S. degree in Computer Science from the University of Sevilla, Spain, in 2006, and the Ph.D. degree in Computer Science and Artificial Intelligence from the University of Granada, Spain, in 2009. He is currently an Assistant Professor in the Department of Computer Science and Numerical Analysis, University of Córdoba, Spain. His current research interests include patter recognition, evolutionary computation, and their applications.

Javier Sánchez-Monedero was born in Córdoba (Spain). He received the B.S.in Computer Science from the University of Granada, Spain, in 2008 and the M.S. in Multimedia Systems from the University of Granada in 2009. In 2013 he obtained the Ph.D. degree on Information and Communication Technologies of the University of Granada. He is working as a researcher with the Department of Mathematics and Engineering (for Department of Quantitative Methods) at the Universidad Loyola Andalucía. His main current research includes machine learning and computational intelligence, as well as their application to real problems.

Athanasia Nikolaou received the B.S. degree in Physics from the University of Athens and the M.Sc. degree in Integrated Climate System Sciences from the University of Hamburg in 2012. She was a Young Graduate Trainee (YGT) in climate dynamics with the European Space Agency, Noordwijk, The Netherlands and currently she is Ph.D. candidate at the German Aerospace Center where she is investigating and modelling magma ocean convection and interaction/feedbacks with secondary atmospheres.



Francisco Fernández-Navarro received the M.Sc. degree in Computer Science from the University of Córdoba, Spain, in 2008, the M.Sc. degree in Artificial Intelligence from the University of Málaga, Spain, in 2009, and the Ph.D. degree in Computer Science and Artificial Intelligence from the University of Málaga in 2011. He was a research fellow in computational management with the European Space Agency, Noordwijk, The Netherlands and currently he is working as an Associate Professor at the Universidad Loyola Andalucía. His current research interests include neural networks, ordinal regression, imbalanced classification, and hybrid algorithms.

César Hervás-Martínez was born in Cuenca, Spain. He received the B.S. degree in Statistics and Operations Research from the "Universidad Complutense", Madrid, Spain, in 1978, and the Ph.D. degree in Mathematics from the University of Seville, Spain, in 1986. He is currently a Professor of Computer Science and Artificial Intelligence in the Department of Computer Science and Numerical Analysis, University of Córdoba, and an Associate Professor in the Department of Mathematics and Engineering (for Department of Quantitative Methods), Universidad Loyola Andalucía. His current research interests include neural networks, evolutionary computation, and the modelling of natural systems.